

| Kurs: Distributed & Mobile Systems |                                     | Code: W6DIST-20 |
|------------------------------------|-------------------------------------|-----------------|
| Studiengang                        | BSc Wirtschaftsinformatik           |                 |
| Studiengruppe                      | BWI-A21                             |                 |
| Dozent/in                          | Walter Rothlin / Frank-Stefan Heinz |                 |
| Leistungsnachweis(e)               | Einzelarbeit mit Fachgespräch       |                 |
| Präsenz                            | Selbststudium                       | Workload        |
| 32 h                               | 40 h                                | 72 h            |

### Lernergebnisse

Die Studierenden

- Sie können eine Client-Server Applikation auf der Basis von TCP/IP Sockets realisieren.
- Sie können verteilte Applikationen konzipieren und entwickeln.
- Sie kennen die grundlegenden Konzepte, Techniken und Methoden, um verteilte und mobile Softwaresysteme zu designen und in Python zu implementieren.
- Sie sind in der Lage, bei Software-Architekturfragen einen entscheidenden Beitrag zur Lösung beizutragen.
- Sie kennen die wichtigsten SOA – Konzepte (SOAP / REST) und können einfache Webservices selbst in Python entwickeln

### Lehrinhalte

1. Client-Server Kommunikation
  - a. XML/JSON
  - b. http-Protokoll
  - c. TCP-IP Socket-Communication
  - d. ➔ XML Service und Client in Python implementieren
2. Verschlüsselung / Authentifizierung
  - a. sym/asym Verschlüsselung – Entschlüsselung
  - b. Signieren – Authentifizieren
  - c. Zertifikate
  - d. ➔ Mit Web-tools Übung machen
3. Eigener Web-Service in Python implementieren
  - a. FLASK / Django
  - b. REST-Service
4. Bestehender Web-Services in Python nutzen
  - a. Unterschied SOAP / REST
  - b. SOAP call von Python aus (gegen JAVA EJB)
  - c. REST-Service call and JSON processing
  - d. Python module **request**
  - e. ➔ Verschiedenen Request-Methoden in Python implementieren
5. Auftrag für Leistungsnachweis
6. 2-Tier Application (Python Frontend / DB as Backend)
  - a. DB-Connection
  - b. Calling an SQL Statement (DML)
  - c. Processing Result-Set
  - d. Calling Stored-Procedure
  - e. Transactions
7. Mobile Entwicklung und Marketing. Ein Rundgang (Mit Frank-Stefan Heinz)
8. Message Queues und IoT Anwendung in Python

### Lehr- und Lernmethoden

- Stoffvermittlung Frontalunterricht, Übungen lösen im Präsenzunterricht und Selbststudium
- **Formative Lernkontrolle:** Zur Vorbereitung auf den Leistungsnachweis müssen die Studierenden zu Fragestellungen entsprechende Lösungen in Python implementieren und die Resultate selbständig überprüfen. Als Entwicklungs-Umgebung wird PyCharm verwendet.

### Leistungsnachweis

Eine Applikation anhand von vorgegebenen Requirements nach den Clean-Code Regeln in Einzelarbeit vor Ort entwickeln. Es sind alle Unterlagen, Beispielcode oder Google erlaubt. Es muss eine Ehrenwörtliche Erklärung abgegeben werden, dass die Arbeit Alleine ohne fremde Personen-Hilfe erstellt wurde.

Geprüft wird der Code anhand der funktionalen und Qualitäts-Kriterien, welche im abgegebenen Prüfungsdokument festgelegt werden. Anhand eines Fachgespräches wird überprüft, ob der Student den Code und die darunter liegenden theoretischen Grundlagen verstanden hat. (Gewicht: 100%)

### Lehrmittel

#### Verpflichtend

Inhalt vom Module Web- & N-tier Applications verstanden. Python-Kurse bestanden und beherrscht seine Entwicklungsumgebung.

**Semesterprogramm****1) Client-Server Kommunikation**

|        |                |
|--------|----------------|
| Datum  | Di 12.3.2024   |
| Zeit   | 17:30 – 20:45  |
| Dozent | Walter Rothlin |

Vorbereitung

- OSI Schichten Modell
- XML und XML Processing
- Python String-Methoden (split, join,...), List and Dicts anwenden können

Inhalte

- http-Protokoll und URL
- Die drei Grundoperationen (Validierung, Transformation und Data-Access) mit einem XML-Dokument in Python implementieren
- JSON Datenstrukturen
- SOAP Services in Python nutzen
- Zustandslose / Zustandsbehaftet (Session)
- Erweitern der TCP/IP Socket Applikation als Service Call
- Ein eigener Client und Server in Python implementieren, welcher über eine Socket-Communication Meldungen (Daten/Aktionen) austauscht.

Nachbereitung

- Testen und fertigstellen der Applikation

**2) Verschlüsselung / Authentication**

|        |                |
|--------|----------------|
| Datum  | Di 2.4.2024    |
| Zeit   | 17:30 – 20:45  |
| Dozent | Walter Rothlin |

Vorbereitung

- Hash-Funktionen (Verlustbehaftete Kompression)

Inhalte

- Codieren / Verschleiern / Tarnen / Verschlüsseln
- Symmetrische Verschlüsselung (Word Dokument Passwort schützen, Shifter)
- Asymmetrische Verschlüsselung (Private / Public-Key)
- Authentifizierung (Signieren und Unterschrift prüfen)
- Zertifikate
- SSL / HTTPS / SFTP

Nachbearbeitung

- Shifter (Mit Hintertüre) in Python realisieren

| 3) Eigener Web-Service in Python implementieren |                |
|---|----------------|
| Datum   | Di 30.4.2024   |
| Zeit  | 17:30 – 20:45  |
| Dozent  | Walter Rothlin |

#### Vorbereitung

- Unterschied FLASK / DJANGO
- Unterschied Common Gateway Interface (CGI) / Web Server Gateway Interface (WSGI)

#### Inhalt:

- FLASK

#### Nachbearbeitung:

- Eigener REST-Service in FLASK implementieren

| 4) Bestehender Web-Services in Python nutzen |                |
|--|----------------|
| Datum  | Di 21.5.2024   |
| Zeit   | 17:30 – 20:45  |
| Dozent                                       | Walter Rothlin |

#### Vorbereitung

Web-Applikationen und die darunterliegenden Services (API) anhand eines Beispiels kennen

- Applikation: <https://openweathermap.org/>
- Service API: <https://openweathermap.org/api>

#### Inhalte

- Web-Service WS (SOAP) in Java: Konzept verstehen und von Python aus nutzen
- API eines public REST-Service (Google, Open-Weather, Map.geo.admin,...) analysieren und in Python nutzen
- Python **request** Modul (unterschiedlichen Request-Methoden) anwenden
- Parameterübergabe in Put/Get

#### Nachbereitung

- Applikation in Python implementieren, welche einen *open* WEB-Service nutzt

| 5) Leistungsnachweis |                |
|----------------------|----------------|
| Datum                | Di 4.6.2024    |
| Zeit                 | 17:30 – 20:45  |
| Dozent               | Walter Rothlin |

#### Inhalte

- **Abgabe der Aufgabenstellung und Kriterien für den Leistungsnachweis**
- **Design und Implementation eines REST-Services**

| 6) 2-Tier Application (Python Frontend / DB as Backend) |                |
|---|----------------|
| Datum   | Di 11.6.2024   |
| Zeit  | 17:30 – 20:45  |
| Dozent  | Walter Rothlin |

#### Vorbereitung

- DB-Anbindung in Python (Siehe Database Module)

#### Inhalt:

- SQL-Abfragen auf Views mit Functions von Python aus
- Python calls Stored-Procedure
- Processing Result-Set

#### Nachbearbeitung

- GUI-Application für Adress-DB entwickeln
- **Parallel dazu Fachgespräche zur eigenen Arbeit (pro Person max 10')**

| 7) Mobile Entwicklung und Marketing. Ein Rundgang |                    |
|---|--------------------|
| Datum   | Di 18.6.2024       |
| Zeit  | 17:30 – 20:45      |
| Dozent  | Frank-Stefan Heinz |

#### Vorbereitung

#### Inhalte

- Mobile Entwicklung und Marketing. Ein Rundgang

#### Nachbearbeitung

| 8) Message Queues and Service Orchestration in Python |                |
|---|----------------|
| Datum   | Di 2.7.2024    |
| Zeit  | 17:30 – 20:45  |
| Dozent  | Walter Rothlin |

#### Vorbereitung

- Funktion und Aufgabe eines Message Brokers erklären können

#### Inhalte

- Message oriented programming in Python mit RabbitMQ
- Web-Services Orchestration (Compose services to a new service)
- IoT Anwendung selber entwickeln (inkl. QR-Code)
- Beispiel Smart Home
- **Parallel dazu Fachgespräche zur eigenen Arbeit (pro Person max 10')**